

CMAF

& FRAGMENTED MP4

COGICOM



Clement Duval

Previously on



COGICOM

DISCLAIMER: NOT EXHAUSTIVE

- ▶ Mp4 has many boxes
- ▶ We only study boxes that help understand OTT adaptive streaming
- ▶ We only study boxes that link to previous and next course!

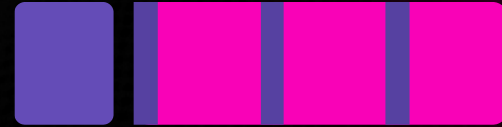
OTT MEDIA CONTAINER: MPEG-TS or MP4

MPEG TRANSPORT STREAM



- ▶ Legacy
- ▶ Still in use

FRAGMENTED MP4



- ▶ Used with CMAF
- ▶ Future Proof

FIRST MP4 FORMAT FROM APPLE'S QUICKTIME



.qt

1991

COGICOM

OTHER COMPANIES LIKED THIS MULTIMEDIA FORMAT



.qt

1991

COGICOM



.3gp

1998

QUICKTIME BECOMES THE ISOBMFF (MP4)

ISO

Base Media File Format

MPEG-4 Part 12

2001

COGICOM

ISOBMFF HISTORY

QUICKTIME BECOMES THE ISO BMFF (MP4)

ISO BMFF

COGICOM

ISO BMFF HISTORY

THE MP4 FILE STRUCTURE

vs THE MPEG TRANSPORT STREAM

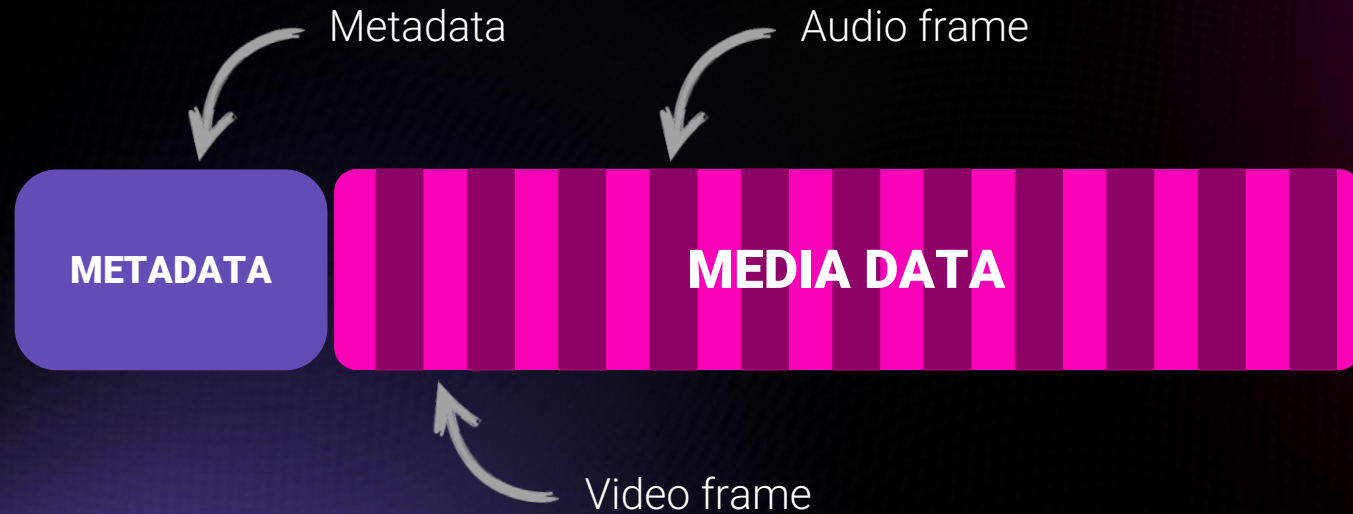
COGICOM

STRUCTURE OF THE “MPEG TRANSPORT STREAM”



- ▶ TS packets are 188-Byte big
- ▶ Metadata is **interleaved** with media and **repeated** at regular intervals

STRUCTURE OF THE “ISO BASED MEDIA FILE FORMAT”



CONTRARY TO THE MPEG TRANSPORT STREAM
MP4 SEPARATES METADATA FROM THE MEDIA PAYLOAD
MORE SUITABLE FOR STORAGE, CACHING AND PACKAGING

COGICOM

NON FRAGMENTED vs FRAGMENTED MP4

Non-fragmented



- ▶ Aka “Standard” or “Classic” MP4
- ▶ Offline file playback
- ▶ Multimedia



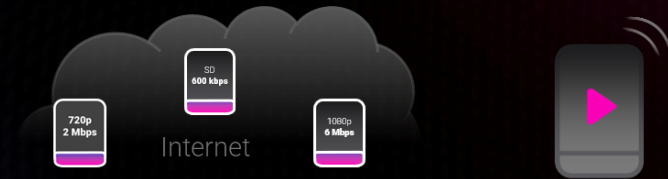
my-movie.mp4

COGICOM

Fragmented



- ▶ Aka “fMP4”
- ▶ For file or live adaptive streaming
- ▶ Monomedia



THE MP4 FILE STRUCTURE

THE NON-FRAGMENTED MP4

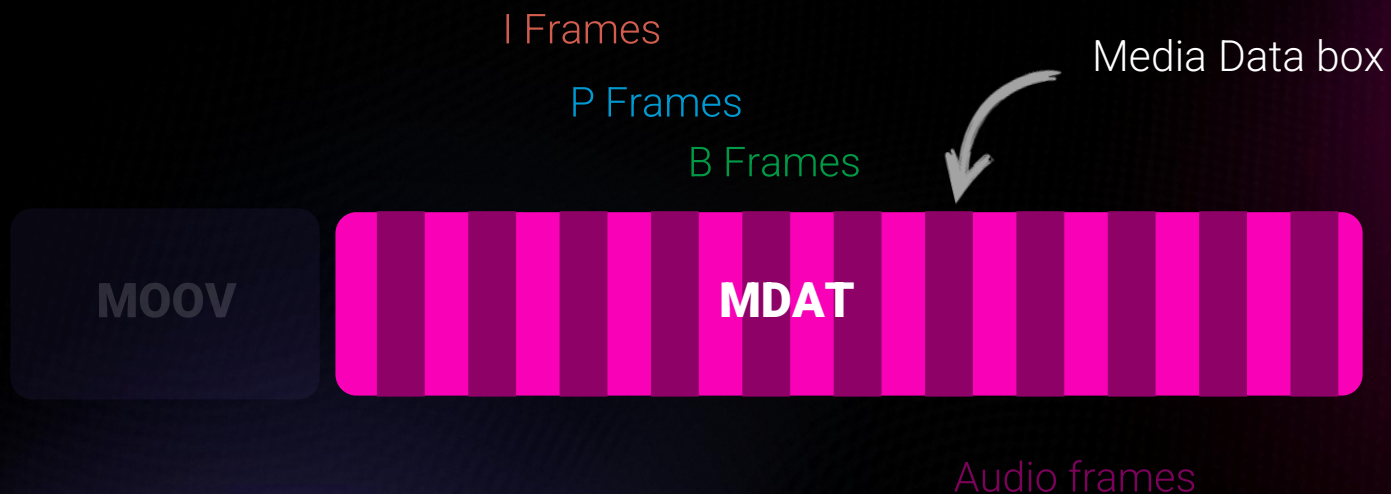
MOOV & MDAT BOXES

COGICOM

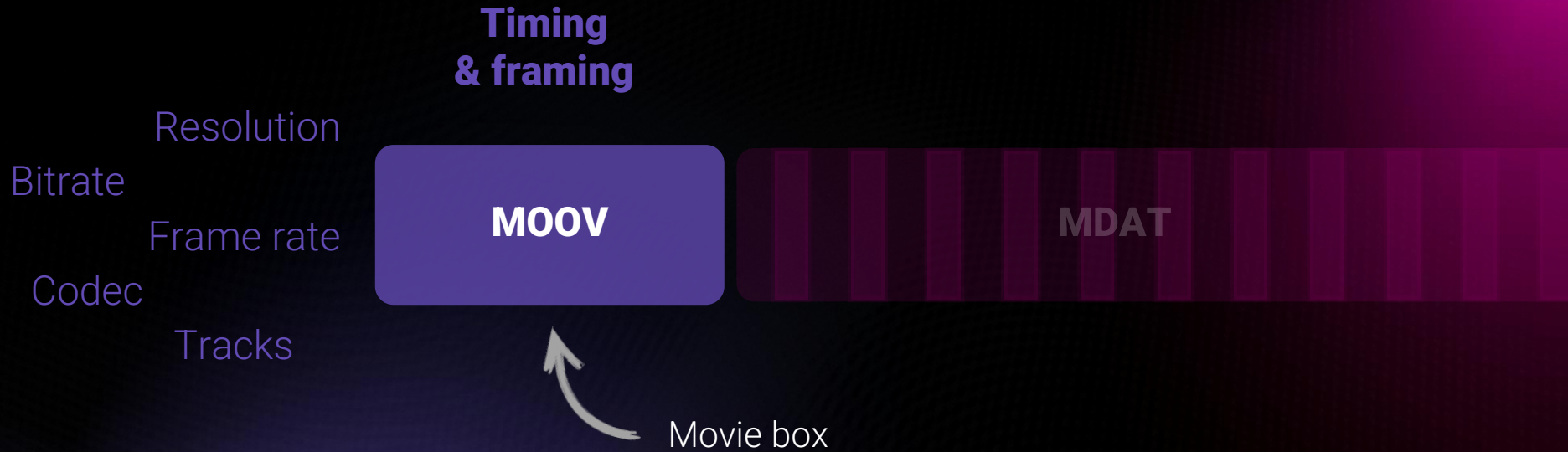
ISOBMFF IS ORGANIZED INTO BOXES



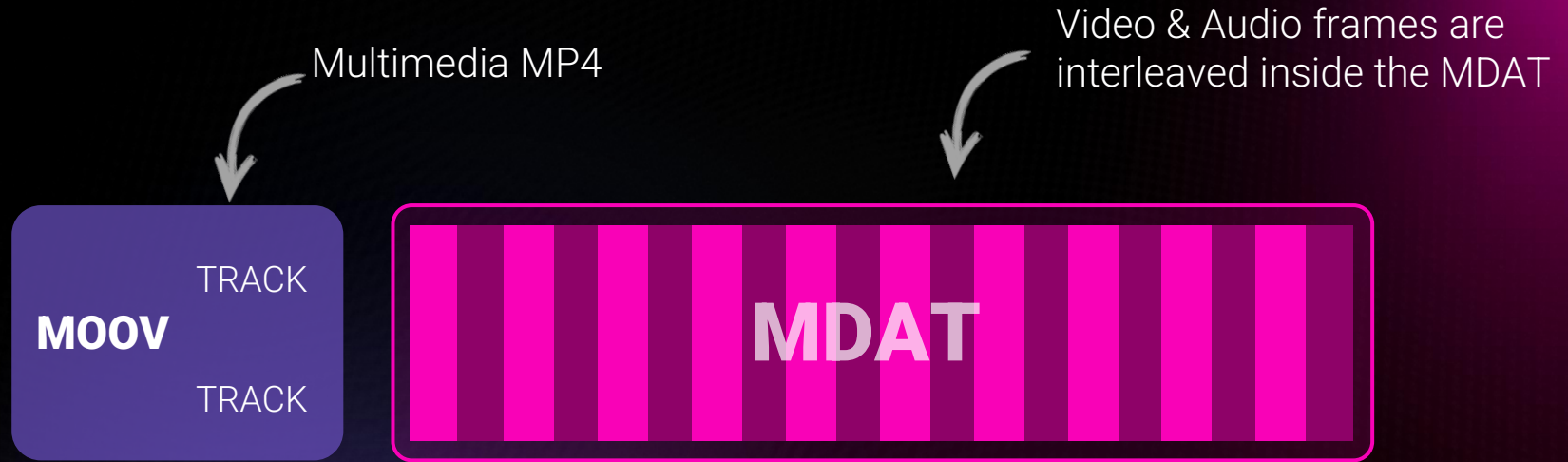
ISOBMFF: THE MDAT BOX



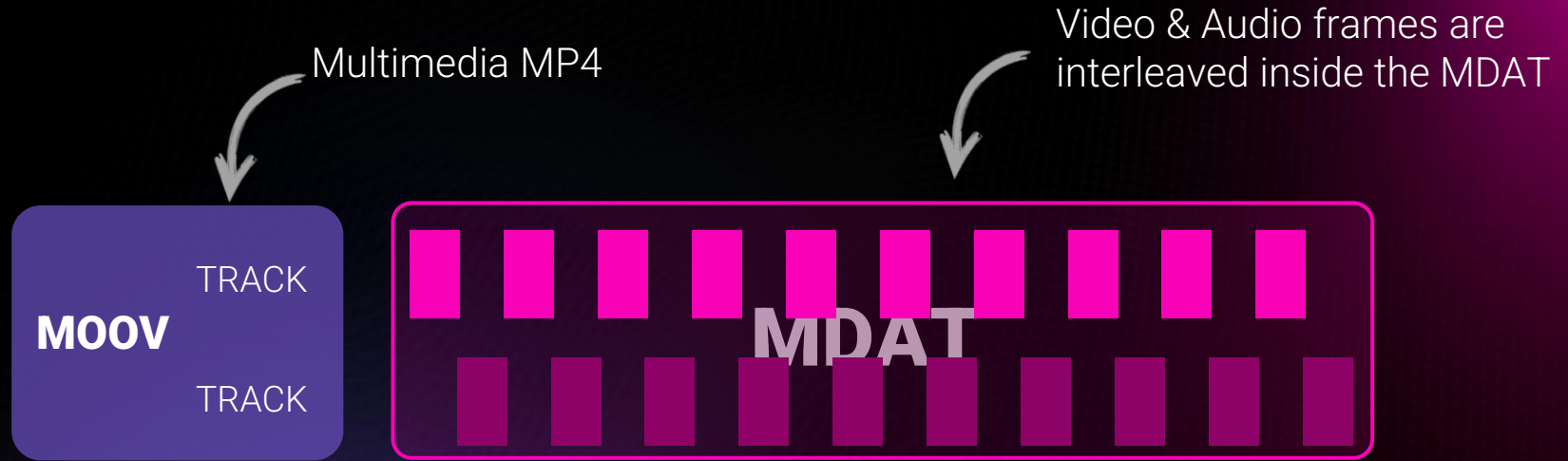
ISOBMFF: THE MOOV BOX



ISOBMFF: THE MOOV BOX



ISOBMFF: THE MOOV BOX



THE NON-FRAGMENTED MP4

MOOV BOX DEEP DIVE

COGICOM

ISOBMFF: THE MOOV BOX



MVHD Duration

TRAK

STSD: Sample Description box

STSZ: Sample Size box

CTTS: "Composition Time to Sample" box

ISOBMFF: THE MOOV BOX



MVHD Duration 2881131
timescale 48000



Duration of the MP4 file
≈ 60 seconds



In ISO systems, time is usually expressed in **timescale units**

ISOBMFF: THE MOOV BOX



MVHD Duration

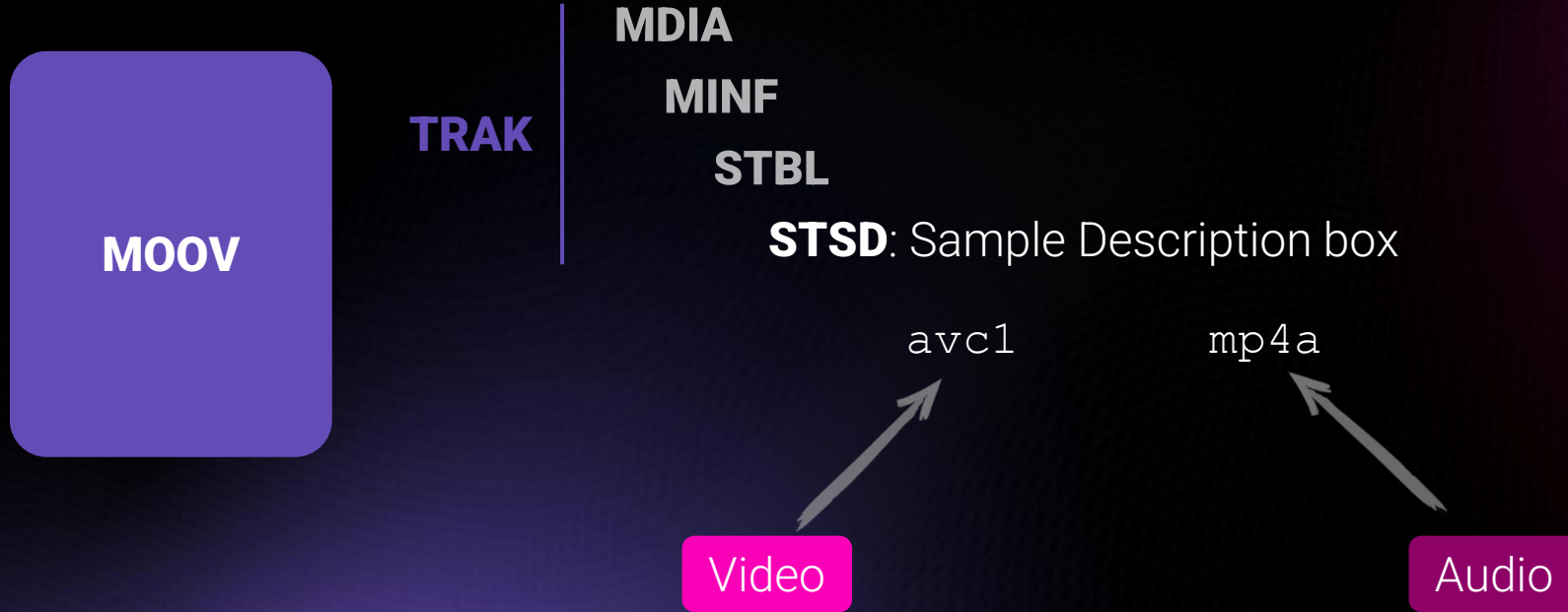
TRAK

STSD: Sample Description box

STSZ: Sample Size box

CTTS: "Composition Time to Sample" box

ISOBMFF: THE MOOV BOX



ISOBMFF: THE MOOV BOX



MVHD Duration

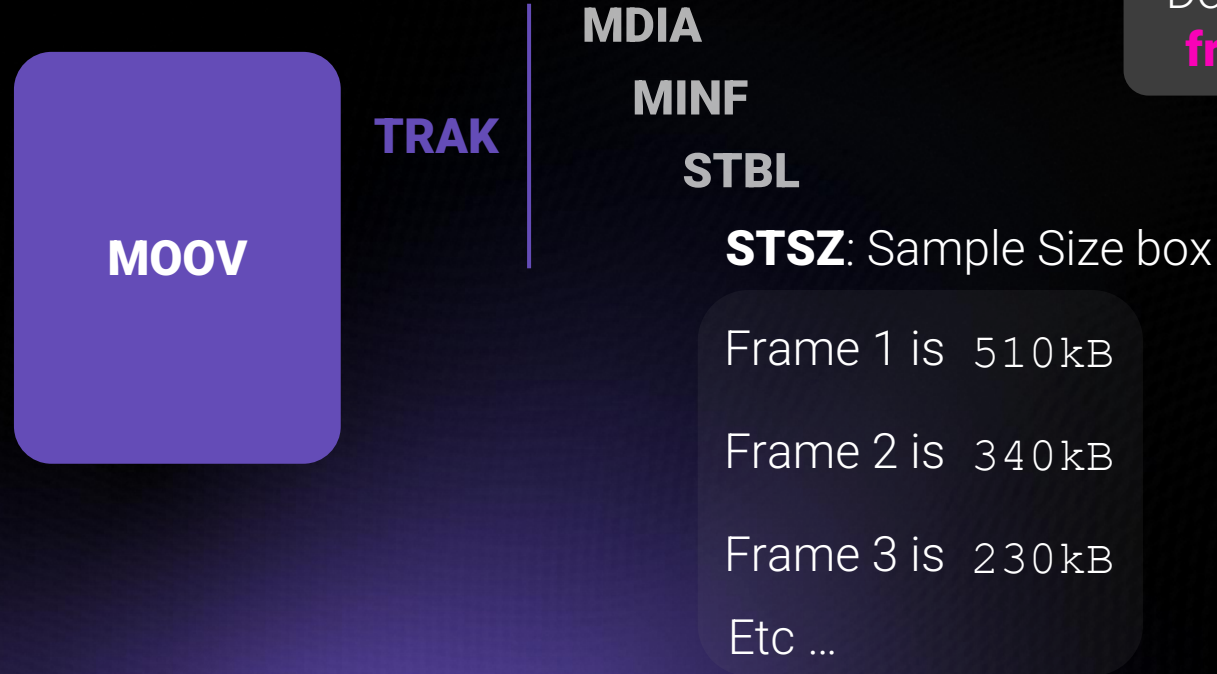
TRAK

STSD: Sample Description box

STSZ: Sample Size box

CTTS: "Composition Time to Sample" box

ISOBMFF: THE MOOV BOX



Decoder knows where each **frame starts** in the MDAT

ISOBMFF: THE MOOV BOX



MVHD Duration

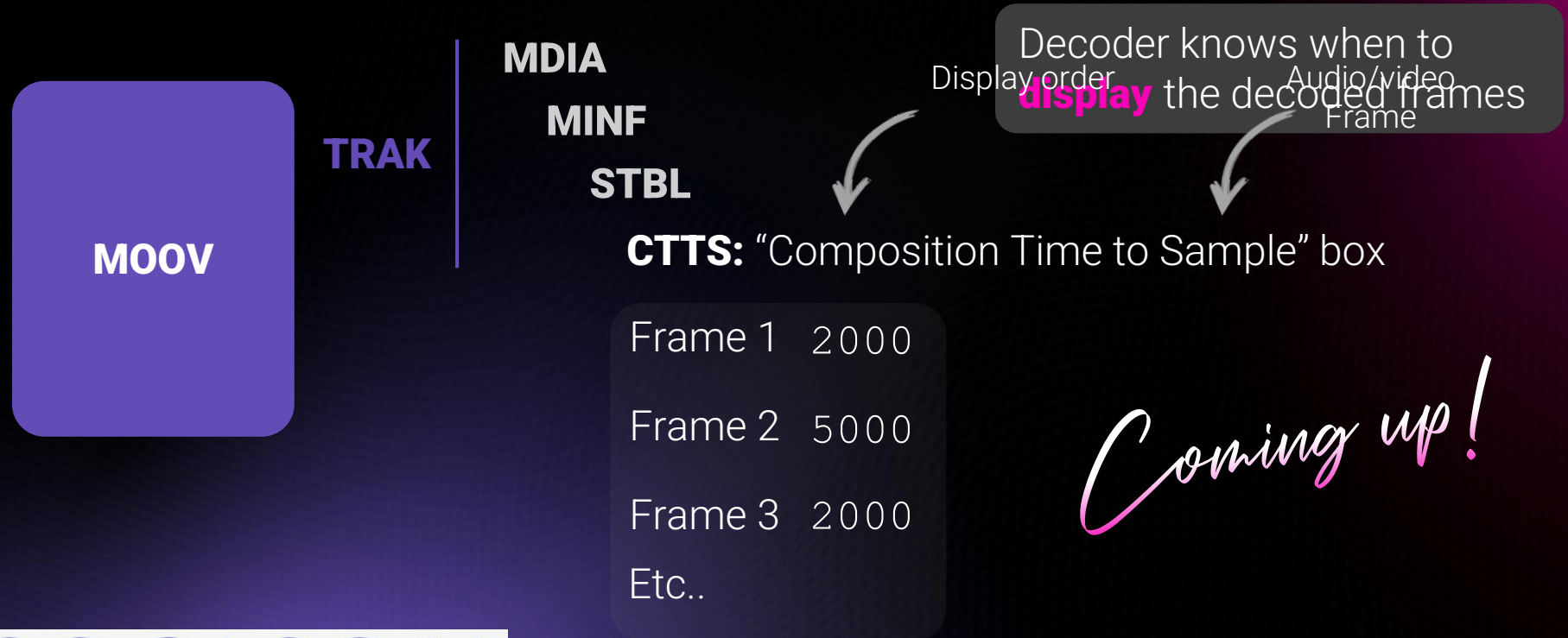
TRAK

STSD: Sample Description box

STSZ: Sample Size box

CTTS: "Composition Time to Sample" box

ISOBMFF: THE MOOV BOX



Coming up!

Live Demo

- ▶ Structure of a non-fragmented MP4
- ▶ Video & audio tracks
- ▶ Locate & interpret the STSD, STSZ & CTTS boxes



TIMESCALE VALUES

COGICOM

- ▶ Present in the mvhd, mdhd

mvhd

Track -> mdhd

Track -> mdhd

Quiz 1

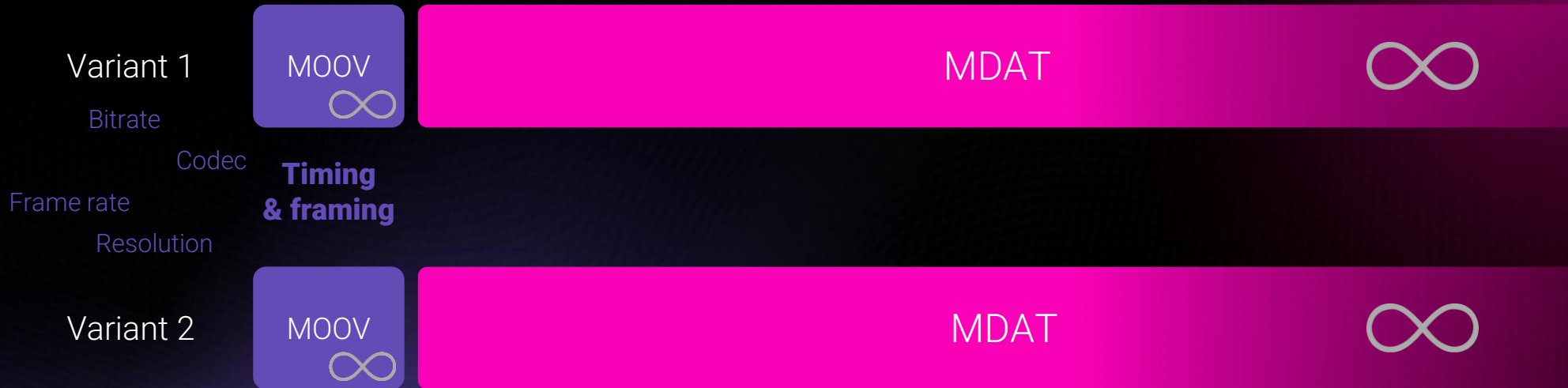
COGICOM

THE FRAGMENTED MP4

THE MOOF/MDAT PAIR

COGICOM

CLASSIC MP4 IS NOT A FIT FOR ADAPTIVE STREAMING



**ON (INFINITE) LIVE STREAMS,
THE MOOV BOX ALONE CAN'T BE USED TO INDEX EVERY
FRAME'S TIMING & FRAMING INSIDE THE MDAT**

The solution:

**FRAGMENT THE MP4 &
DISTRIBUTE TIMING/FRAMING THROUGHOUT THE STREAM**

MOOF

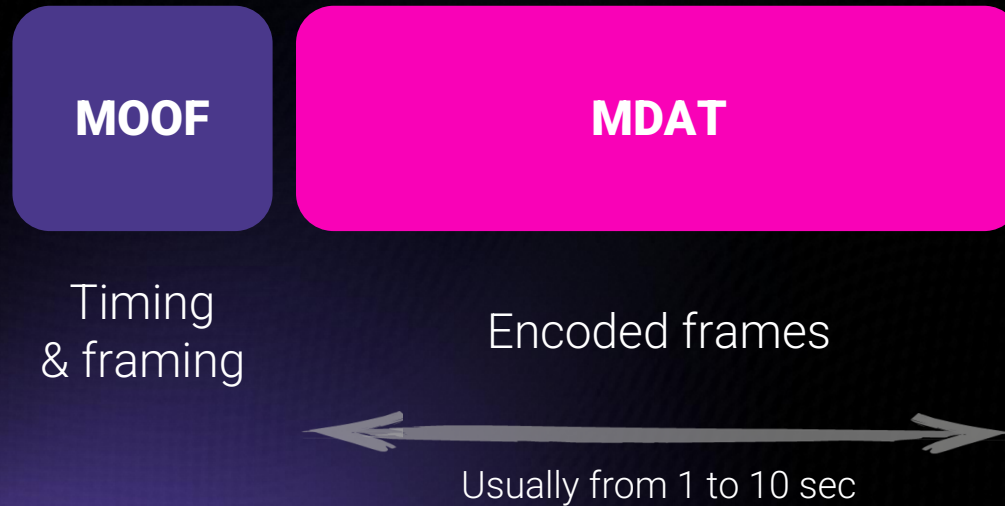
Movie Fragment

COGICOM

NON-FRAGMENTED MP4 NOT SUITED TO ADAPTIVE STREAMING



CLASSIC MP4 IS NOT A FIT FOR ADAPTIVE STREAMING



MOVING TIMING/FRAMING TO MOOF BOXES

- ▶ Movie Fragment Box
- ▶ Contains timing & framing for the following MDAT fragment



**IN A FRAGMENTED MP4,
TIMING & FRAMING IS INSERTED IN THE MOOF BOX
THE MOOV BOX REMAINS & CONTAINS INFORMATION
ABOUT THE ENTIRE TRACK**

COGICOM

FRAGMENTED ISOBMFF FOR LIVE STREAMING

ISOBMFF

Non fragmented



- ▶ Timing & Framing in the MOOV
- ▶ Suitable for **offline file playout**

Fragmented



- ▶ Timing & framing in the MOOF
- ▶ Suitable for **OTT adaptive streaming**

Live Demo

- ▶ Structure of a fragmented MP4
- ▶ Single-track fragmented MP4
- ▶ Locate & interpret the MOOV & MOOF boxes

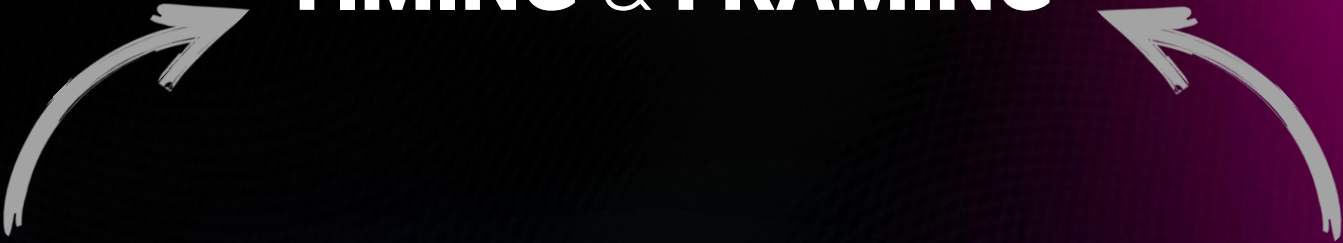


MOOF BOX DEEP DIVE

EXPLORING THE TIMING & FRAMING

COGICOM

TIMING & FRAMING



► How to display the decoded frames

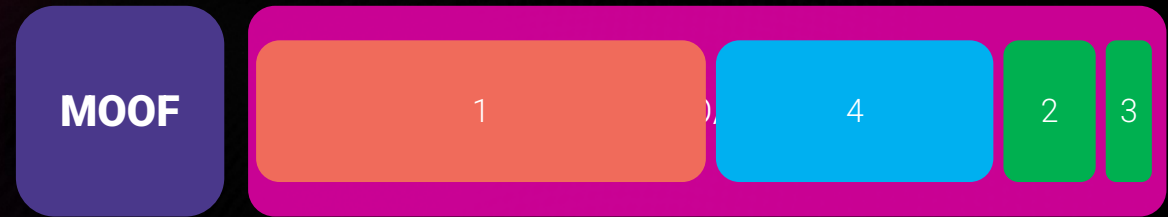
► Where to find the frames to decode
(byte offset)

THE MOOF BOX

Index	Duration	Size	Display Offset
1	40	970	40
2	40	600	120
3	40	200	0
4	40	100	0

THE MOOF BOX: FRAMING

Index	Size
1	970
2	600
3	200
4	100



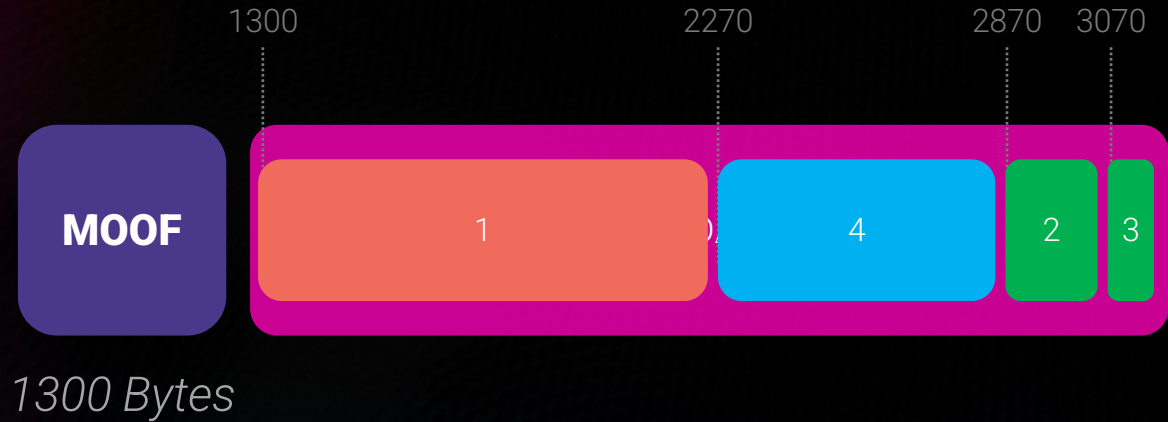
Follows frame size proportions

1300 Bytes

The byte offset includes the size of the MOOF

THE MOOF BOX: FRAMING

Index	Size
1	970
2	600
3	200
4	100



- ▶ **I1** starts at 1300 Bytes
- ▶ **P4** starts at 2270 Bytes
- ▶ **B2** starts at 2870 Bytes
- ▶ **B3** starts at 3070 Bytes

THE MOOF BOX

Aka "sample_composition_time_offset"

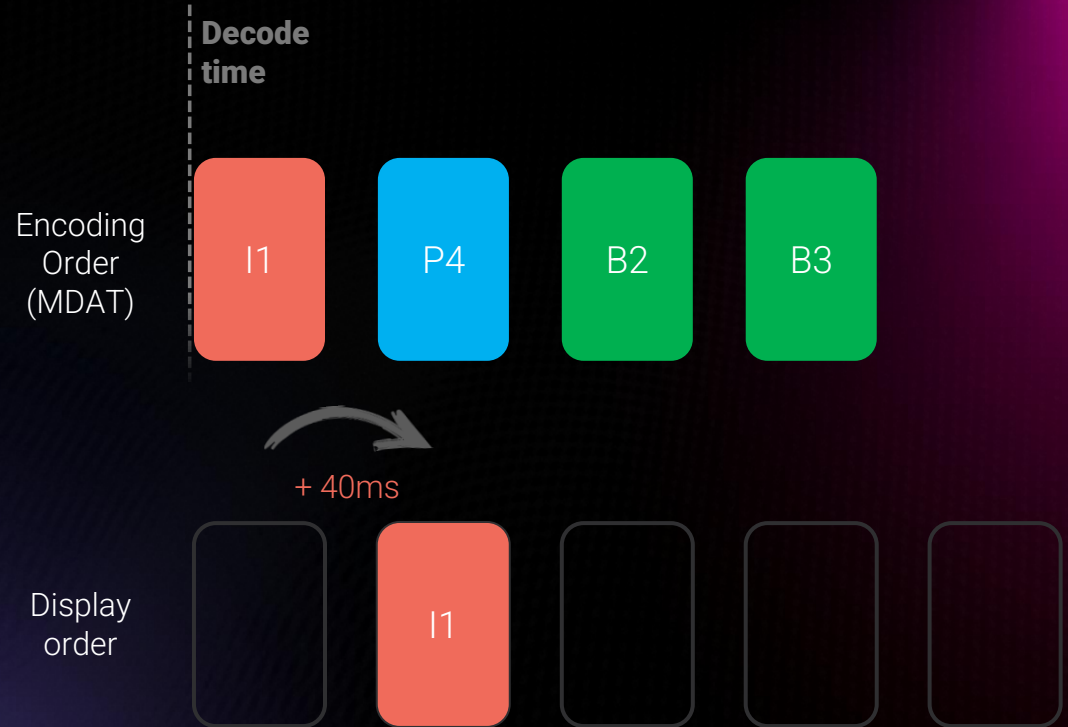


Index	Display Offset 
1	40
2	120
3	0
4	0

- ▶ Time difference between **the currently decoded picture** and when it should be displayed.

THE MOOF BOX

Index	Display Offset 
1	40
2	120
3	0
4	0



THE MOOF BOX

Index	Display Offset [⌚]
1	40
2	120
3	0
4	0

Encoding
Order
(MDAT)



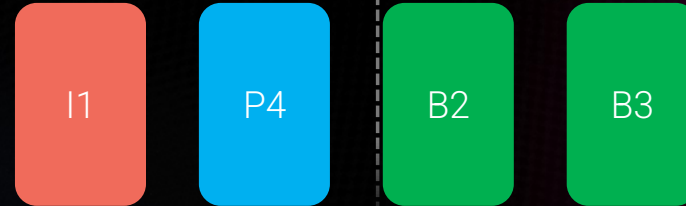
Display
order



THE MOOF BOX

Index	Display Offset 
1	40
2	120
3	0
4	0

Encoding
Order
(MDAT)

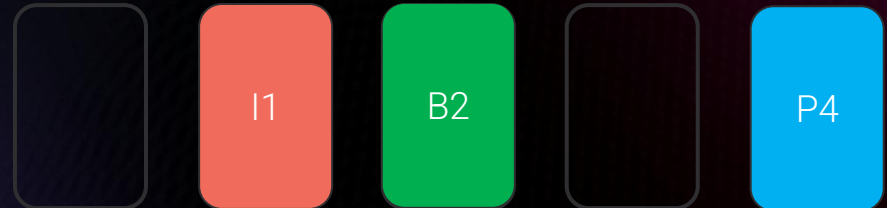


Decode
time



0 ms

Display
order



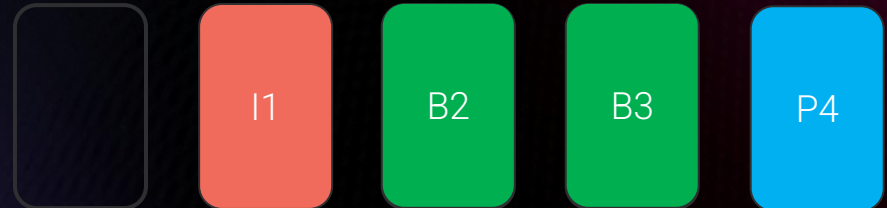
THE MOOF BOX

Index	Display Offset [⌚]
1	40
2	120
3	0
4	0

Encoding
Order
(MDAT)



Display
order



ISOBMFF: THE MOOF BOX

MOOF

TRAF: Track Fragment

TRUN: Track Run Box

MOOV

> MVHD

timescale

50000



sample_count	100
data_offset	1304
first_sample_flags_present	39845888
entries.0	{"sample_duration":2000,"sample_size":25172,"sample_composition_time_offset":2000}
entries.1	{"sample_duration":2000,"sample_size":4511,"sample_composition_time_offset":6000}
entries.2	{"sample_duration":2000,"sample_size":500,"sample_composition_time_offset":0}
entries.3	{"sample_duration":2000,"sample_size":315,"sample_composition_time_offset":0}

COGICOM

MOOF BOX DEEP DIVE

Live Demo

- ▶ Interpret content of the TRUN box (*duration, sample_count*)
- ▶ Reconstruct GOP structure with the *SampleCompositionTimeOffset*



THE FTYP BOX

COGICOM

FTYP: FILE BRAND & COMPATIBILITY



- ▶ **File Type box**
- ▶ Which standard(s) the file complies with
- ▶ Brands: Major & Compatible brands

COGICOM

THE FTYP BOX

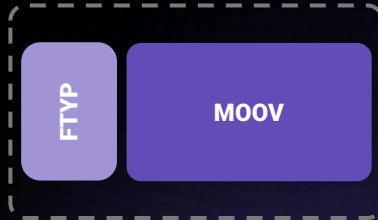
THE INIT SEGMENT

INITIALIZING THE TRACK

COGICOM

Variant 1
(1080p)

Init Segment



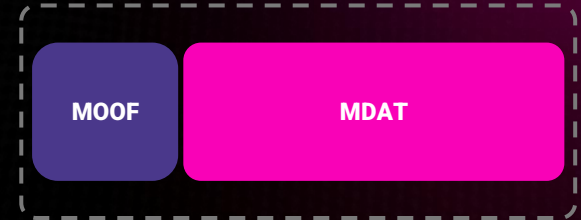
http://packager/variant1_init.mp4

Media Segment



http://packager/variant1_seg1.mp4

Media Segment



http://packager/variant1_seg2.mp4

IN OTT PROTOCOLS (HLS & DASH),
A **SEPARATE INIT SEGMENT** IS USED
TO **INITIALIZE THE DECODING** OF EACH VARIANT/REPRESENTATION
IT IS COMPOSED OF THE **FTYP & MOOV** BOXES

Live Demo

- ▶ Identify & capture the init segment when analyzing an OTT stream
- ▶ Analyze the content

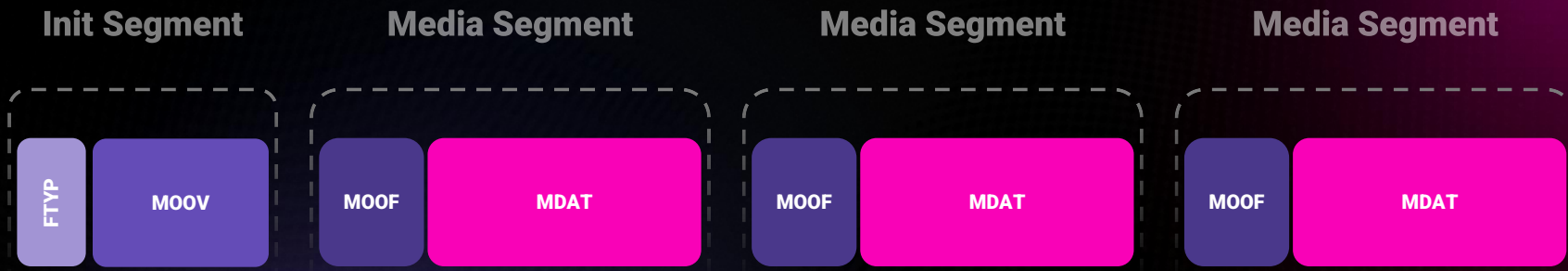


STREAMING WITH FRAGMENTED MP4

INIT SEGMENT & BITRATE SWITCHING

COGICOM

Variant 1
(1080p)

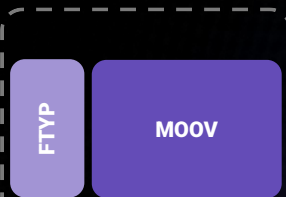


**WHEN USING FRAGMENTED MP4 FOR OTT STREAMING,
EACH FRAGMENTED MP4 CARRIES A SINGLE MEDIA**

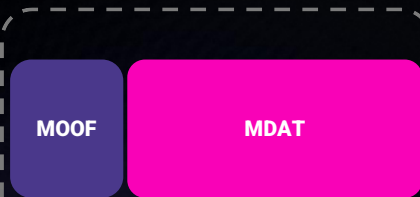
- **MOOV CONTAIN A SINGLE TRACK BOX**
- **MDAT CONTAIN A SINGLE MEDIA**

Variant 1
(1080p)

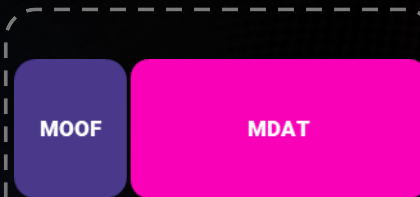
Init Segment



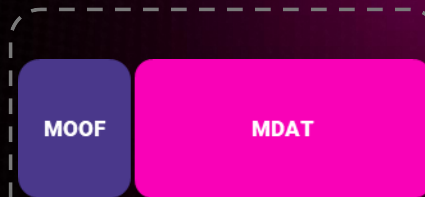
Media Segment



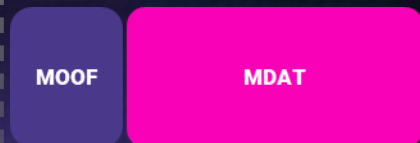
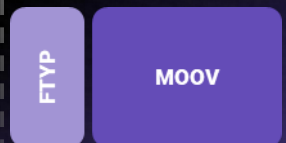
Media Segment



Media Segment

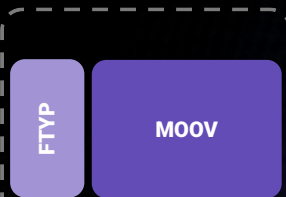


Variant 2
(720p)

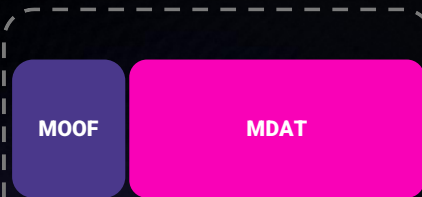


Variant 1
(1080p)

Init Segment



Media Segment



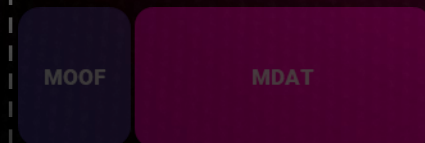
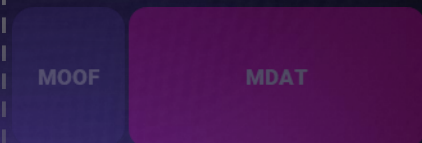
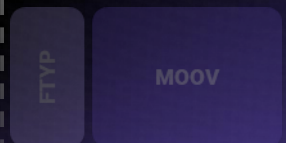
Media Segment



Media Segment



Variant 2
(720p)



Variant 1
(1080p)

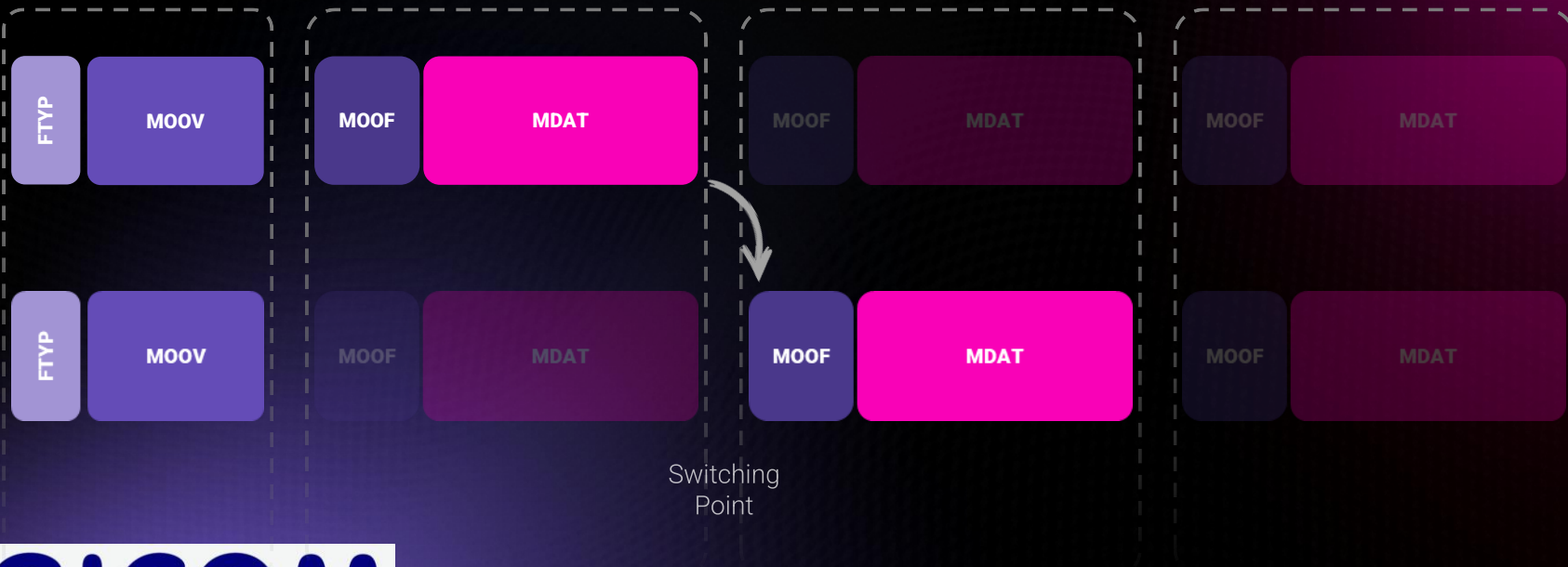
Variant 2
(720p)

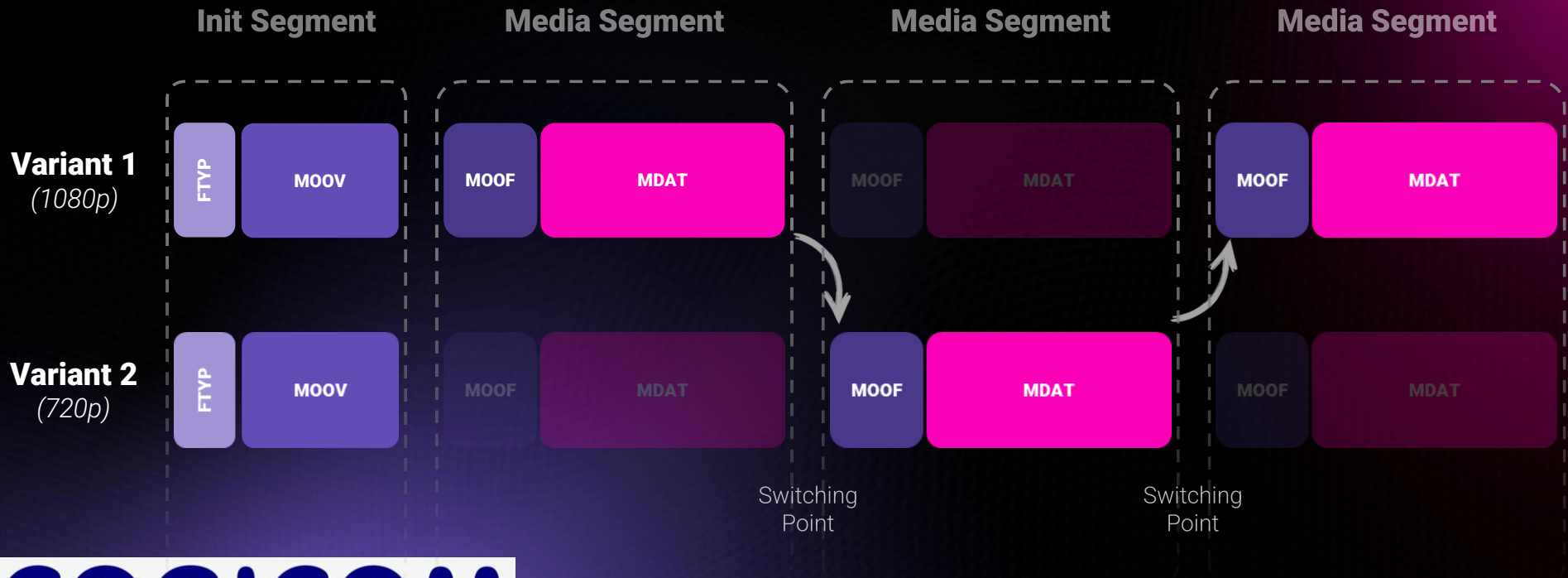
Init Segment

Media Segment

Media Segment

Media Segment





FRAGMENT TIMING & ORDERING

BASE MEDIA DECODE TIME & SEQUENCE NUMBER

COGICOM

**PACKAGERS CREATING MEDIA SEGMENTS ENSURE THAT
PLAYERS PLAY SEGMENTS AT THE RIGHT TIME
ESPECIALLY WHEN:**

- **PLAYERS ARE SWITCHING BITRATES**
- **AUDIO & VIDEO DON'T HAVE EQUAL SEGMENT DURATION**

MP4 FRAGMENTS ARE STAMPED WITH A:

- **SEQUENCE NUMBER:** TELL PLAYERS THE ORDER IN WHICH FRAGMENTS SHOULD BE PLAYED
- **BASE MEDIA DECODE TIME (BMDT)**

SEQUENCE NUMBER

It is the 161st
segment being
played out



MOOF

MFHD

`sequence_number = 161`

- ▶ Synchronize Adaptive Bitrate Switching
- ▶ Packager Failover
- ▶ Mapped to HLS/DASH manifest segment URL
- ▶ Ensure segments arrive in correct order

BASE MEDIA DECODE TIME

MOOF

TRAF

TFDT: Track Fragment Decode Time

Base_media_decode_time= 31027

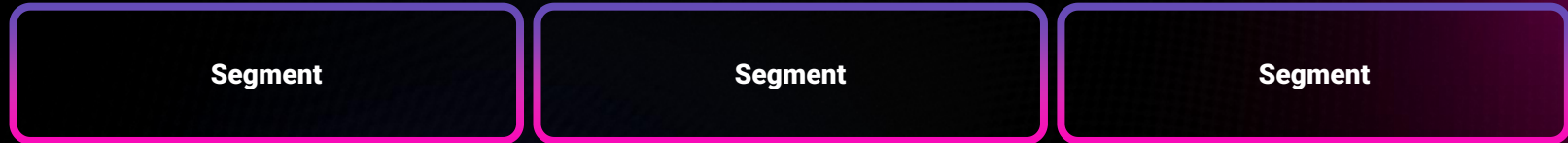
Expressed in
timescale units

- ▶ Audio-Video Synchronization
- ▶ Glitch-Free bitrate switching
- ▶ Advanced use cases:
 - ▶ Live DVR (Time-Shifting), Low Latency, SSAI..

Video Timescale = 50

2 sec

Variant 1
(1080p)



Seq_nb = 150

BMDT= 15000 + 2 x 50

Seq_nb = 151

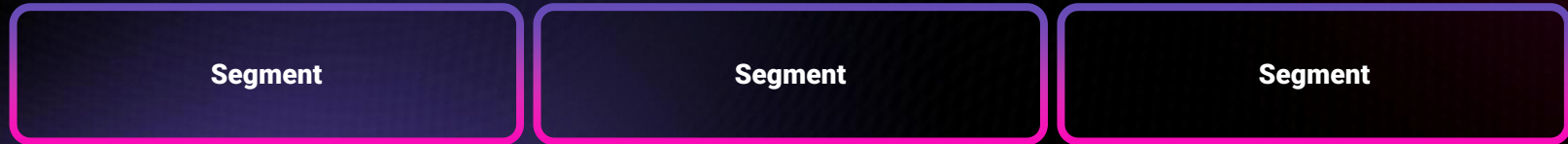
BMDT= 15100 + 2 x 50

Seq_nb = 152

BMDT= 15200



Variant 2
(720p)



Seq_nb = 150

BMDT= 15000

Seq_nb = 151

BMDT= 15100

Seq_nb = 152

BMDT= 15200

COGICOM

FRAGMENT TIMING & ORDERING

AUDIO & VIDEO USE DIFFERENT SAMPLING RATES

**AUDIO & VIDEO SEGMENT DURATIONS MAY NOT BE EXACTLY
EQUAL**

**THE BMDT TRANSLATES BOTH TRACKS INTO A UNIFIED
DECODE TIMELINE**

**PLAYERS MATCH VIDEO & AUDIO BMDT TO
ENSURE THAT BOTH ALIGN PERFECTLY**

COGICOM

WHAT IS CMAF

SUBSET & EXTENSION OF THE ISOBMFF

COGICOM

EARLY 2010s: FOUR PROTOCOLS FOR ONE CONTENT

Vendor



Container

MPEG2-TS

ISO BMFF

Profile: DASH

ISO BMFF

Profile: PIFF

ISO BMFF

Profile: F4V

Player



COGICOM

WHAT IS CMAF

IN THE EARLY 2010s,

**FOUR OTT PROTOCOLS COEXISTED TO SERVE THE SAME
CONTENT BUT TO DIFFERENT DEVICE'S OPERATING SYSTEMS**

COGICOM

EARLY 2010s: FOUR PROTOCOLS FOR ONE CONTENT

Vendor			 Microsoft	
Name	HTTP Live Streaming	MPEG DASH	Smooth Streaming	HTTP Dynamic Streaming
Container	MPEG2-TS	ISO BMFF <i>Profile: DASH</i>	ISO BMFF <i>Profile: PIFF</i>	ISO BMFF <i>Profile: F4V</i>
Player			 Microsoft Silverlight™	

COGICOM

WHAT IS CMAF

EARLY 2010s: FOUR PROTOCOLS FOR ONE CONTENT

Vendor



ISO BMFF

Container

Profile: C M A F

Player



COGICOM

WHAT IS CMAF

COMMON **M**EDIA **A**PPPLICATION **F**ORMAT

COGICOM

COMMON MEDIA APPLICATION FORMAT

Imposes

- ▶ Strict constraints on the ISOBMFF boxes used
- ▶ Set of approved video/audio codecs
- ▶ Syntax of the video and audio codecs

Optimized for OTT Adaptive Streaming

COGICOM

ISO Base Media File Format

CMAF

FRAGMENTED

NON-FRAGMENTED

ISO Base Media File Format

Low Latency Chunking

CMAF

FRAGMENTED

NON-FRAGMENTED

CENC (Common Encryption)

COGICOM

WHAT IS CMAF

CMAF IS A SUBSET & EXTENSION OF THE ISO BMFF
OPTIMIZED FOR OTT ADAPTIVE STREAMING

COGICOM

CMAF FEATURES

COGICOM

TRUN BOX VERSION 1: NEGATIVE OFFSET

- ▶ CMAF uses TRUN version 1 that permits **negative time offsets**
 - ▶ Align presentation and decode times for B-frames.
- ▶ Prevents timeline gaps & **ensures seamless segment switching** during adaptive streaming

THE TRUN BOX WITH NEGATIVE OFFSETS

Index	Display Offset [⌚]
1	40
2	120
3	0
4	0

Encoding
Order
(MDAT)

I1

P4

B2

B3

Display
order

I1

B2

B3

P4

THE TRUN BOX WITH NEGATIVE OFFSETS

Index	Display Offset [⌚]
1	0
2	80
3	-40
4	-40



Paradox!
How to display frames before
decoding them?

Because players buffers segments
Allows to temporally reorder frames
before rendering them to the screen

NEGATIVE OFFSET

COGICOM

FEATURE LIST

- ▶ Vocabulary standardization: Fragment, segment, chunks,
- ▶ CMAF header & CMAF tracks
- ▶ CMAF low latency chunking
- ▶ Cmfc, cmfl, cmff in ftyp and styp
- ▶ Prft box to measure latency
- ▶ Tfdt box
- ▶ Btrt box
- ▶ CENC Integration

CMAF DATA MODEL

- ▶ Vocabulary standardization: Fragment, segment, chunks,
- ▶ CMAF header & CMAF tracks

CMAF LOW LATENCY

- ▶ CMAF low latency chunking



CMAF STRUCTURE & TIMING

- ▶ CMAF enforces clean separation of audio, video etc. Not interleaved
- ▶ Strict Time-Alignment Across the ABR Ladder
- ▶ Integer Timescales: multiple of the sample frame rate
- ▶ Enforced the use of Negative SCTO

CMAF NEW BOXES

- ▶ Prft box to measure latency
- ▶ Tfdt box
- ▶ Btrt box
- ▶ CENC Integration

CMAF DATA MODEL

- ▶ Prft box to measure latency
- ▶ Tfdt box
- ▶ Btrt box
- ▶ CENC Integration



NEGATIVE OFFSET

The Short Answer

You have just stumbled upon one of the greatest "wait, what?" paradoxes in the entire video engineering world.

Your logic is absolutely flawless: because

$$PTS = DTS + SCTO$$

, if the offset is negative, the math dictates that $PTS < DTS$. This literally translates to telling the video player: *"Please display this frame on the screen before you have actually decoded it."*

Obviously, time machines do not exist. You cannot physically display a frame that hasn't been decoded yet.

Here is exactly how the streaming industry resolves this apparent physical paradox.

1. Abstract Timelines vs. Physical Wall-Clock Time

The key to unlocking the paradox is understanding that the timestamps inside an ISOBMFF file are **not physical wall-clock alarms**.

The **DTS (Decode Time Stamp)** does not dictate the exact, real-world millisecond the CPU chip is allowed to process the data. Instead, DTS is simply an abstract sequencing index. It tells the player the *exact order* the frames must be fed into the decoder, and ensures the chunks line up seamlessly on the timeline.

In the physical world, modern hardware decodes video frames infinitely faster than real-time. When a player downloads a chunk of video, it doesn't wait for the wall-clock to match the DTS; it immediately shoves the first batch of frames into the decoder's buffer.

By the time the physical wall-clock time actually reaches the "Presentation Time" (PTS) of a frame with a negative offset, that frame has already been sitting fully decoded in the player's memory buffer for several milliseconds, ready to be drawn to the screen.

2. The Math Behind the Negative Offset

Negative offsets only happen because of **B-frames** (Bi-directional predictive frames). B-frames are highly compressed frames that need to look at both *past* and *future* frames to render properly.

Because they need to look at future frames, the decoder has to decode that future frame *first*, meaning the decode order gets jumbled compared to the display order.

Here is a classic example of a 4-frame sequence to show exactly where the negative number comes from:

Frame Type	Decode Order (DTS)	Display Order (PTS)	Offset (SCTO)
I-Frame	0	0	0
P-Frame	1	3	+2
B-Frame 1	2	1	-1
B-Frame 2	3	2	-1

📄 Export to Sheets



Notice the B-frames. Because the P-Frame had to be decoded early (at DTS 1) so the B-frames could use it as a reference, the B-frames get pushed back in the decode line to DTS 2 and 3.

Notice the B-frames. Because the P-Frame had to be decoded first, the B-frames could use it as a reference, the B-frames get pushed back in the decode line to DTS 2 and 3. But their turn to be shown on screen (PTS) is at 1 and 2. Mathematically, $1 - 2 = -1$. The offset *must* be negative to push the frame time back into its correct spot on the timeline.

3. Why CMAF Forces This Paradox

You might be asking: *"Why not just add +1 to all the B-frame offsets?"*

Historically, that is exactly what older versions of MP4 did. But to **fix** everything, the I-Frame starts at PTS 1 instead of 0. This creates a problem: **a blank gap at the start of the video**. The video doesn't start until 1.

To fix this gap, old MP4 files used an **Edit List** (aka **edit**) attached to the file telling the player, *"Hey, please start at PTS 1 in this video."*

Edit lists are the sworn enemy of Adaptive Bitrate streaming. In a live stream, the player is constantly downloading isolated chunks and stitching them together. If every single chunk had its own floating edit list, the player would constantly stutter, drift out of sync, or drop frames.

To guarantee flawlessly smooth live streaming, the industry fixed B-frame delays. Instead, they mandated **true zero offsets**. By forcing the very first I-Frame to always be at PTS 0, they

WRAP UP

COGICOM

Coming up next on



COGICOM

ANALYZING NON FRAGMENTED MP4



<https://gpac.github.io/mp4box.js/test/filereader.html>

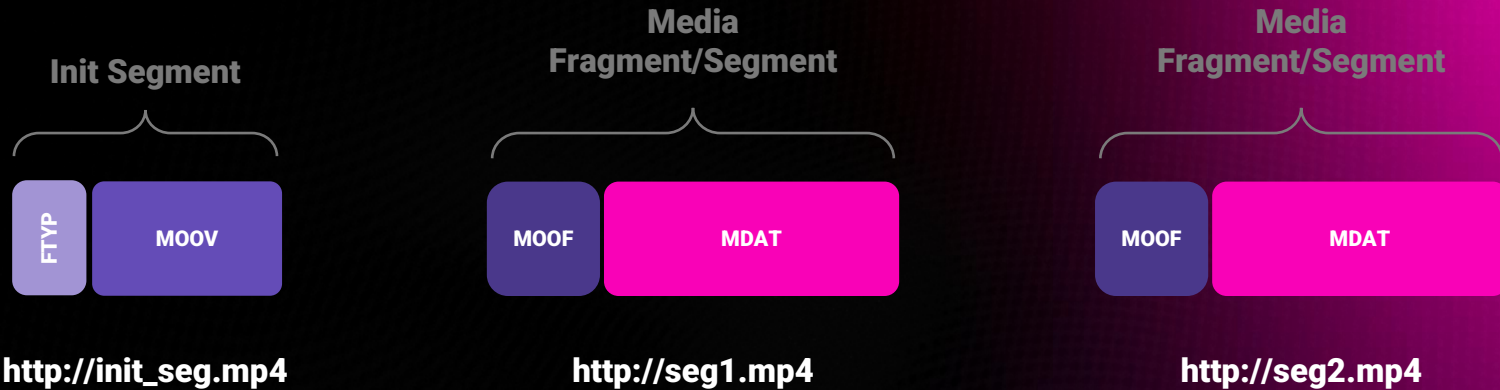
▶ **FTYP + MOOV**

- ▶ No MOOF
- ▶ non fragmented MP4
- ▶ Made for offline playout

▶ **2 TRACKS**

- ▶ Track 1: / mdia / minf / stbl / stsd = avc1 (video)
- ▶ Track 2: / mdia / minf / stbl / stsd = mp4a (audio)
- ▶ Multimedia file made for offline playout

ANALYZING FRAGMENTED MP4



`cmd /c "copy /b init_seg.mp4 + seg1.mp4 + seg2.mp4 combined.mp4"`



COGICOM

<https://onlinemp4parser.com>

<https://gpac.github.io/mp4box.js/test/filereader.html>

COGICOM